CSE 559A: Computer Vision

[credit: danjodon.deviantart.com]

Fall 2017: T-R: 11:30-1pm @ Lopata 101

Instructor: Ayan Chakrabarti (ayan@wustl.edu).
Staff: Abby Stylianou (abby@wustl.edu), Jarett Gross (jarett@wustl.edu)

http://www.cse.wustl.edu/~ayan/courses/cse559a/

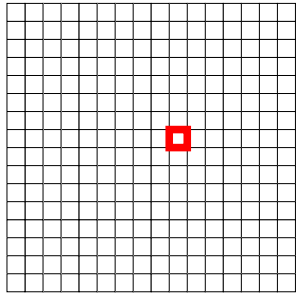
Aug 31, 2017

ADMINISTRIVIA

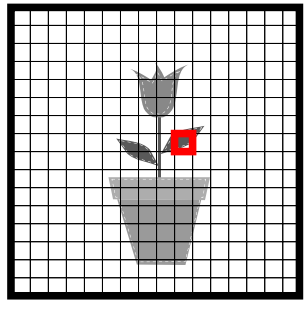
- Everyone needs to fill out survey and join Piazza.
- We strongly encourage you to use Piazza over e-mail for questions (mark only to instructors if you wish).
- Also read Piazza posts !
- Doodle poll to fix office hours was posted Tue. Vote by midnight tonight.
- We'll take a snapshot then, and decide office hours and recitation time.
- Reminder: slides will be posted after class. (Tuesday slides already online)
- Additional Python and Math resources posted on project website. (scroll to end)
- LaTeX resources to be posted soon.
- Everyone said they have a laptop. If you are concerned about compute power for problem sets, e-mail me.

- Everyone on waitlist as of Tue morning was enrolled. Pretty much at limit now.

SENSOR



Image



Sensor Plane

(ignoring color for now)

$$I[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t) \;\; p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y}) \; q \; dx \; dy \right] dt$$

- n_x, n_y are integers indexing pixels in image array.
- $I[n_x, n_y]$ is recorded pixel intensity.
- (x, y) is spatial location
- $(\bar{x}_{n_x}, \bar{y}_{n_y})$ is the "center" spatial location of the pixel / sensor element at $[n_x, n_y]$.

SENSOR

$$I[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t) \;\; p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y}) \; q \; dx \; dy \right] dt$$

- n_x, n_y are integers indexing pixels in image array.
- $I[n_x, n_y]$ is recorded pixel intensity.
- (x, y) is spatial location
- $(\bar{x}_{n_x}, \bar{y}_{n_y})$ is the "center" spatial location of the pixel / sensor element at $[n_x, n_y]$.

- $E(x, y, t)$ is light "power" per unit area incident at location (x, y) on the sensor plane at time t
- $p(x, y)$ is spatial sensitivity of the sensor (might be lower near boundaries, etc.)
- q is quantum efficiency of the sensor (photons/energy to charge/voltage)
- T is the duration of the exposure interval.

SENSOR

$$I[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t) \;\; p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y}) \; q \; dx \; dy \right] dt$$

$I[n_x, n_y]$ is the *ideal unquantized* pixel intensity

SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

$$I \leftarrow I^0$$

SHOT NOISE

- Caused by uncertainty in photon arrival
- Actual number of photons K is a discrete random variable with Poisson distribution
- $P(K = k) = \frac{\lambda^k e^{-\lambda}}{k!}$
- λ is the "expected" number of photons. In our case, $\propto I^0$
- Property of Poisson distribution: Mean and Variance both equal to λ
- Often, shot noise is modeled with additive Gaussian noise with signal dependent variance:

$$I \leftarrow I^0 + \sqrt{I^0}\ \ \epsilon_1$$

where $\epsilon \sim \mathcal{N}(0, 1)$ (Gaussian random noise with mean 0, variance 1).

SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t) \;\; p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y}) \; q \; dx \; dy \right] dt$$

$$I \leftarrow I^0 + \sqrt{I^0} \;\; \epsilon_1$$

AMPLIFICATION & ADDITIVE NOISE

- Signal amplified by gain g before digitization. Based on ISO (higher g for higher ISO).
- Some signal-independent Gaussian noise added before and after amplification.

$$I \leftarrow g \times (I^0 + \sqrt{I^0} \;\; \epsilon_1 \; + \sigma_{2a}\epsilon_{2a}) + \sigma_{2b}\epsilon_{2b}$$

where σ_{2a} and σ_{2b} are parameters (lower for high quality sensors),
and ϵ_1 , ϵ_{2a} , ϵ_{2b} are $\mathcal{N}(0, 1)$ noise variables, all independent.

$$I \leftarrow \underbrace{gI^0}_{\text{Amplified Signal}} + \underbrace{g\sqrt{I^0} \;\; \epsilon_1}_{\text{Amplified Shot Noise}} + \underbrace{\sqrt{\left(g^2\sigma_{2a}^2 + \sigma_{2b}^2\right)} \;\; \epsilon_2}_{\text{Amplified and un-amplified additive noise}}$$

SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

$$I \leftarrow gI^0 + g\sqrt{I^0}\ \ \epsilon_1 + \sqrt{\left(g^2\sigma_{2a}^2 + \sigma_{2b}^2\right)}\ \ \epsilon_2$$

DIGITIZATION

- Final step is rounding and clipping (by an analog to digital converter)

$$I \leftarrow \text{Round}\left(gI^0 + g\sqrt{I^0}\ \ \epsilon_1 + \sqrt{\left(g^2\sigma_{2a}^2 + \sigma_{2b}^2\right)}\ \ \epsilon_2 \right)$$

$$I = \min\left(I_{\max},\ \ \text{Round}\left(gI^0 + g\sqrt{I^0}\ \ \epsilon_1 + \sqrt{\left(g^2\sigma_{2a}^2 + \sigma_{2b}^2\right)}\ \ \epsilon_2 \right) \right)$$

SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

$$I = \min\left(I_{\max},\ \ \text{Round}\left(gI^0 + g\sqrt{I^0}\ \ \epsilon_1 + \sqrt{\left(g^2\sigma_{2a}^2 + \sigma_{2b}^2 \right)}\ \ \epsilon_2 \right) \right)$$

ignoring sensor saturation, dark current, ...

WHY STUDY THIS ?

- To understand the degradation process of noise (if we want to denoise / recover I^0 from I).
- To prevent degradation during capture, because we control exposure time T and ISO / gain g.
- To understand the different trade-offs for loss of information from noise, rounding, and clipping.

SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

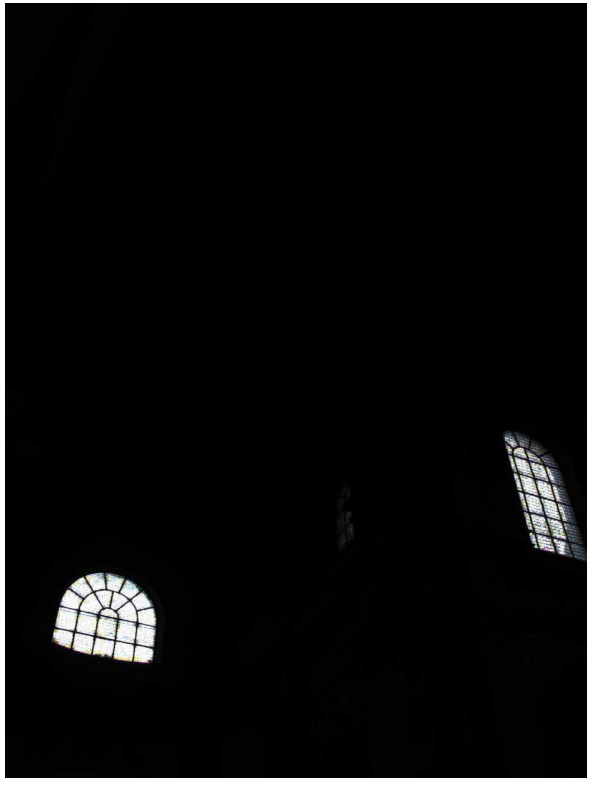
$$I = \min\left(I_{\max},\ \ \text{Round}\left(gI^0 + g\sqrt{I^0}\ \ \epsilon_1 + \sqrt{\left(g^2\sigma_{2a}^2 + \sigma_{2b}^2\right)}\ \ \epsilon_2 \right) \right)$$

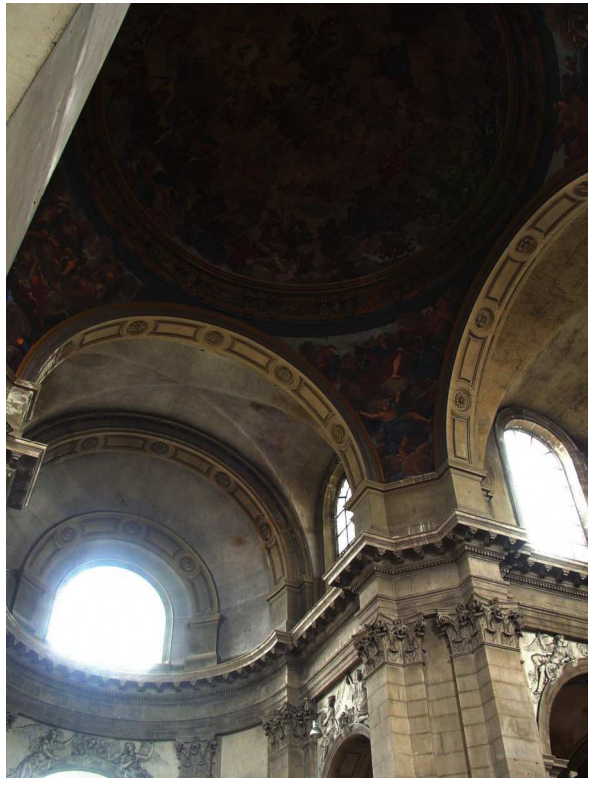
ROUNDING VS CLIPPING

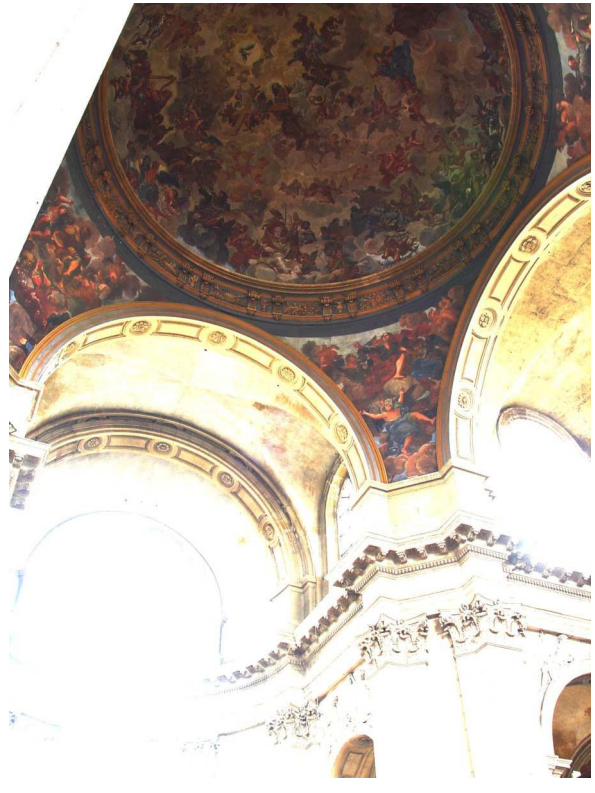
Ignoring noise, what is the optimal g for a given $I^0[n_x, n_y]$?

- Keep g low so that most values of $gI^0[n_x, n_y]$ are below $I_{\max}$.
- But if g is too low, a lot of the variation will get rounded to the same value.

SENSOR



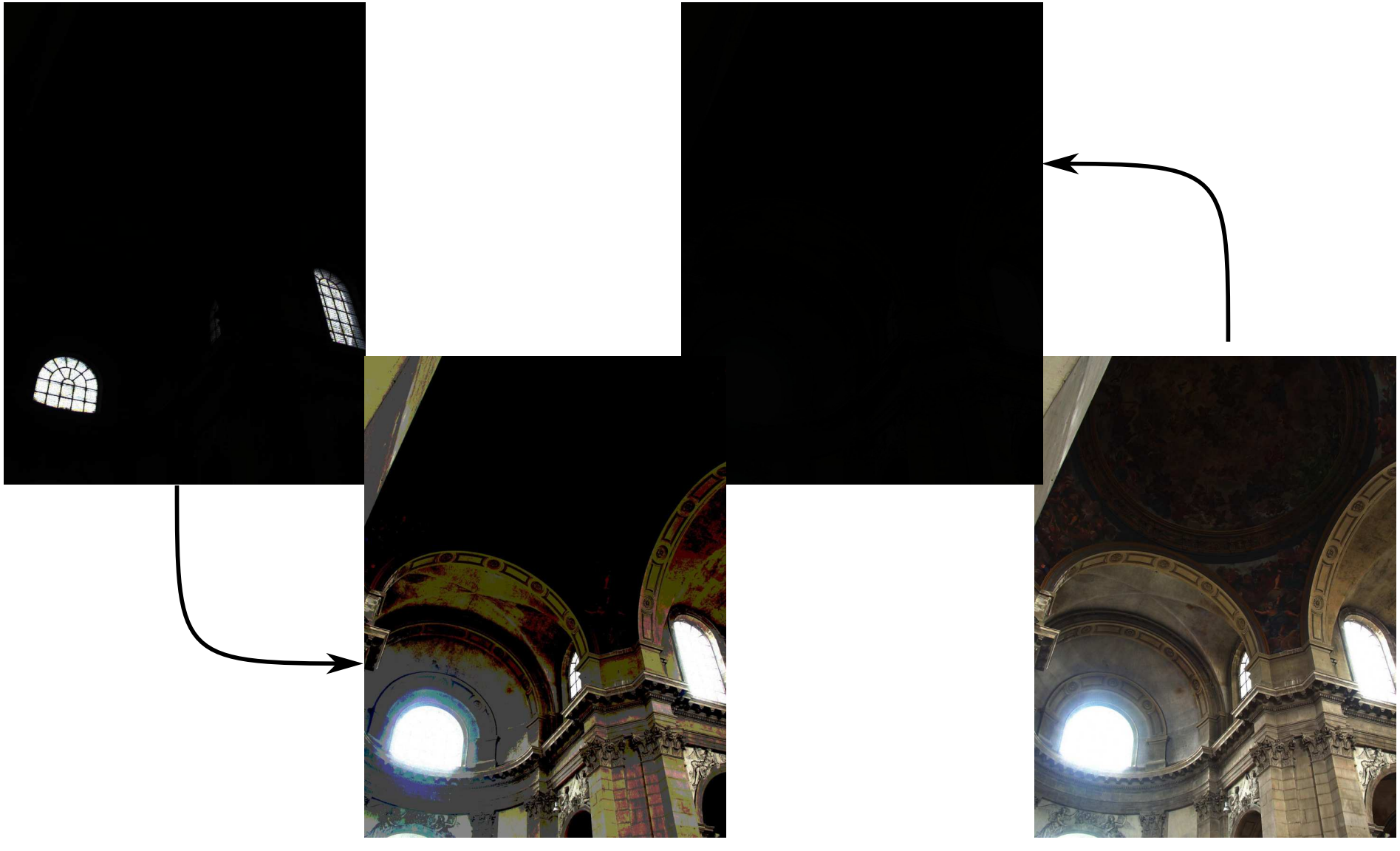




Increasing g

Data from Sam Hasinoff

SENSOR



SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

$$I = \min\left(I_{\max},\ \ \text{Round}\left(gI^0 + g\sqrt{I^0}\ \ \epsilon_1 + \sqrt{\left(g^2\sigma_{2a}^2 + \sigma_{2b}^2\right)}\ \ \epsilon_2 \right) \right)$$

LIGHT VS AMPLIFICATION

Say we have chosen the optimal target values for the product gI^0. Is it better:

- To have a higher g and lower magnitude I^0
- **To have a lower g and higher magnitude I^0**
- Depends, based on σ_{2a}, σ_{2b}

Additional Reading (if interested):

S. Hasinoff, F. Durand, W.T. Freeman, "Noise-Optimal Capture for High Dynamic Range Photography," CVPR 2010.

SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

So how do we increase I^0 ?

- Better sensors (higher q)
- Larger sensor elements: $p(\cdot, \cdot) > 0$ over a larger area.

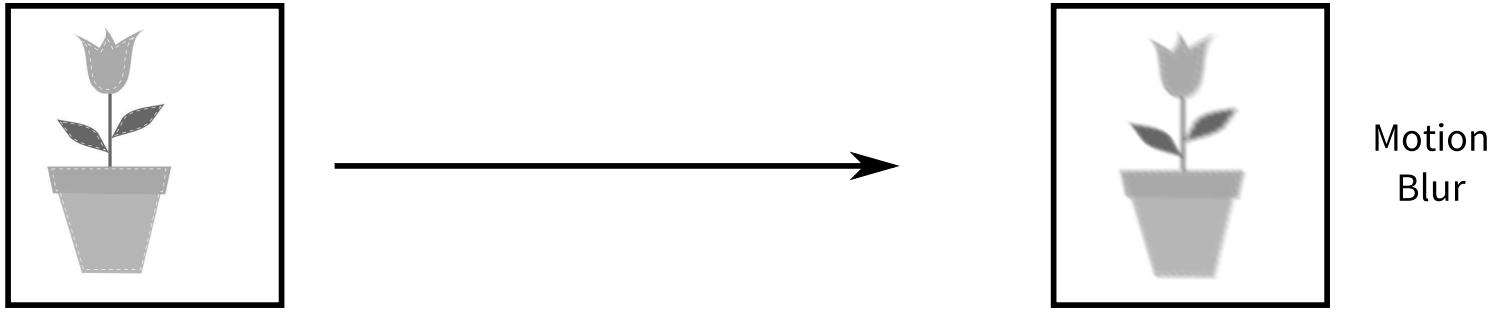
  But we've gone the other way: cameras stuff more 'megapixels' in smaller form factors.

SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

Increase exposure time T ?

- If scene is static and camera is stationary:
  - $E(x, y, t)$ doesn't change with $t \Rightarrow I^0 \propto T$
- If scene is moving ...



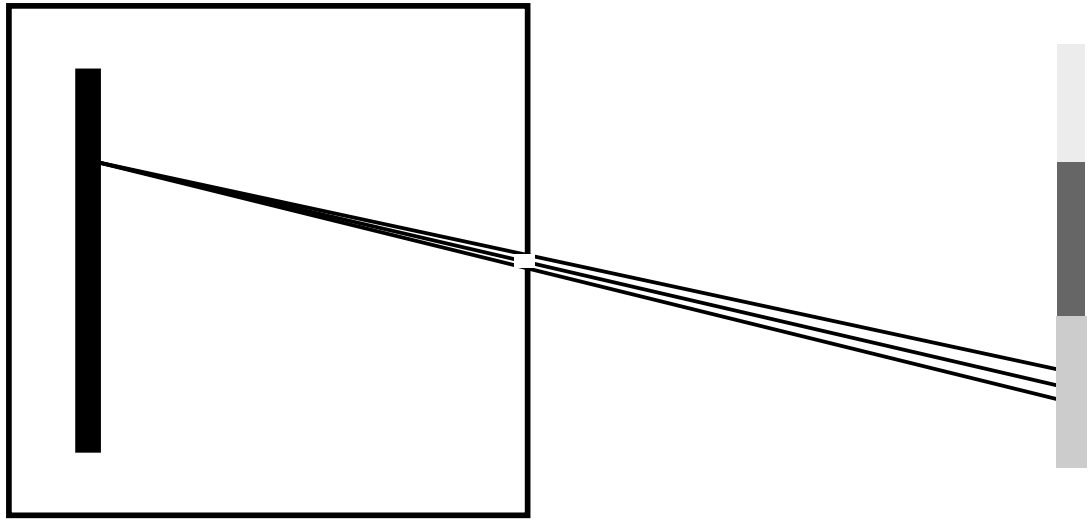
SENSOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t) \;\; p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y}) \; q \; dx \; dy \right] dt$$

Increase $E(x, y, t)$ itself. How ?

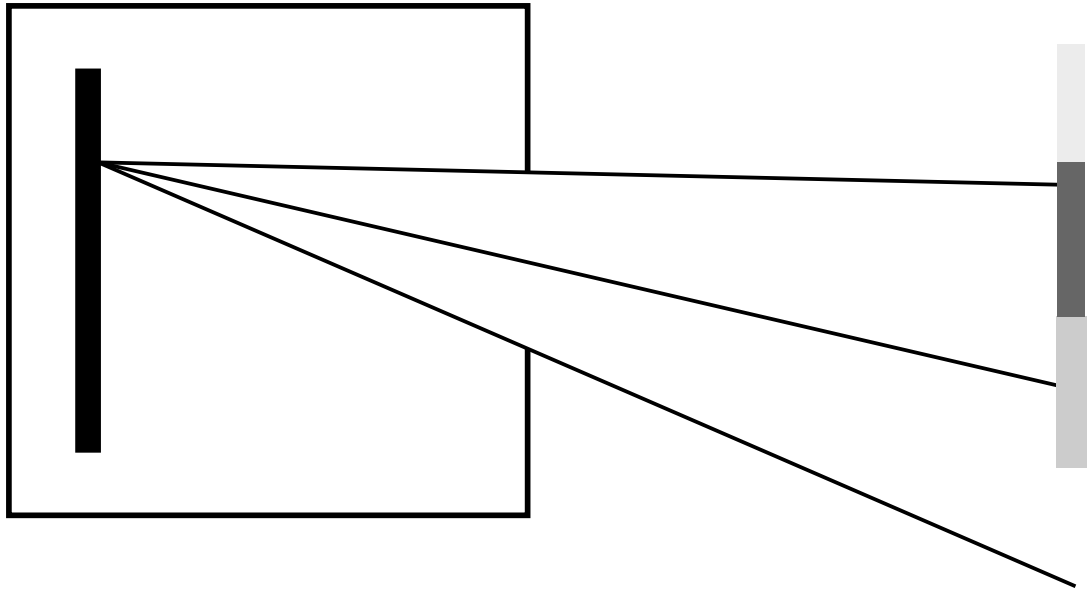
- Take pictures outdoors, or under brighter lights.
- Don't use a pinhole camera !

LENSES



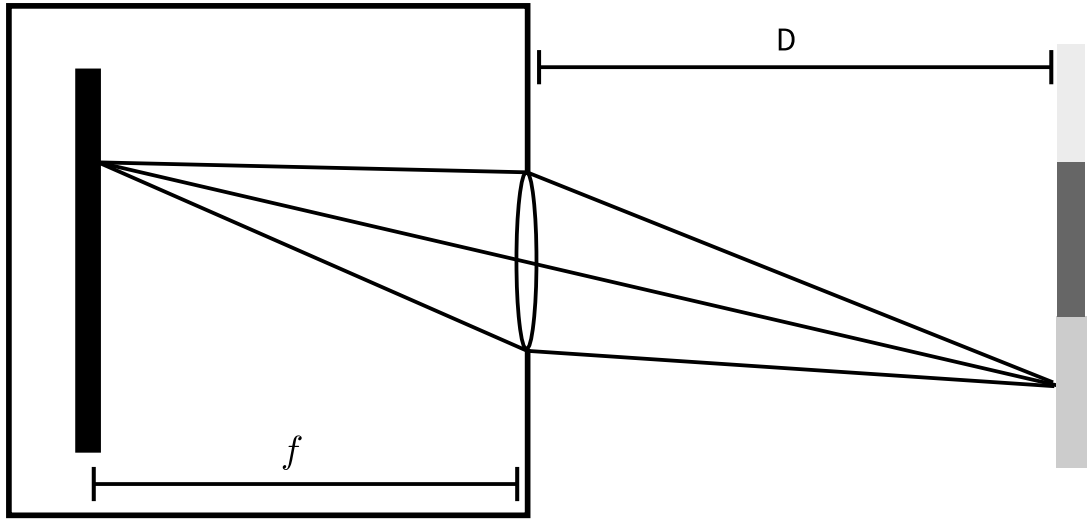
Small Pinhole: Focused beam, very little light.

LENSES



Large Pinhole: More light, larger 'circle of confusion'

LENSES



$$\frac{1}{D} + \frac{1}{f} = \frac{1}{\tilde{f}}$$

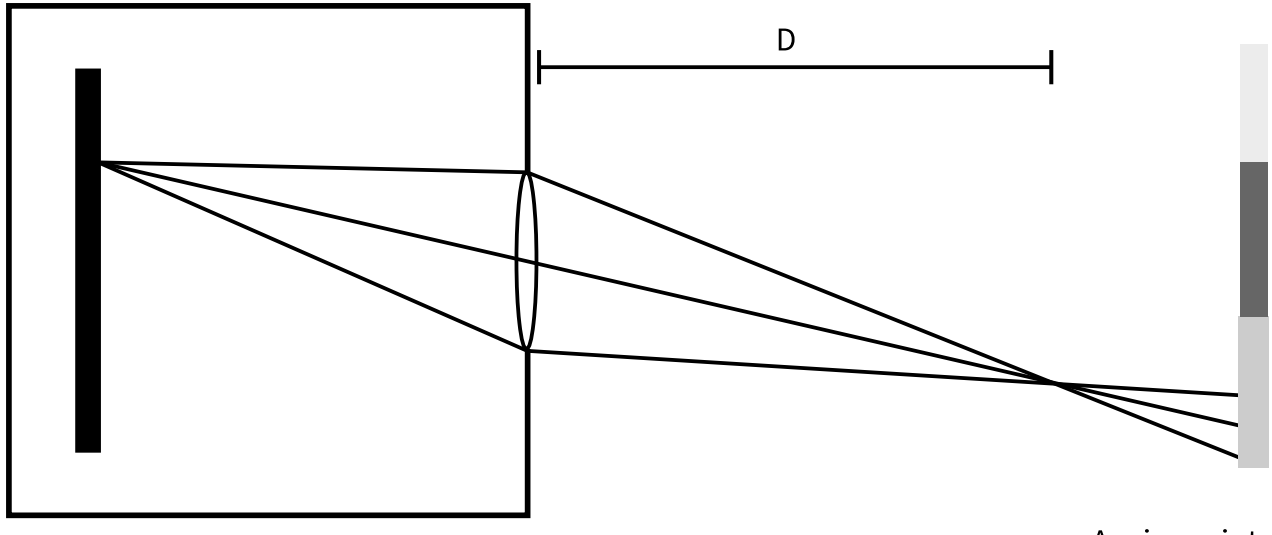
Focal length of Lens

Distance of Sensor to Aperture

Thin lens model (approximation)

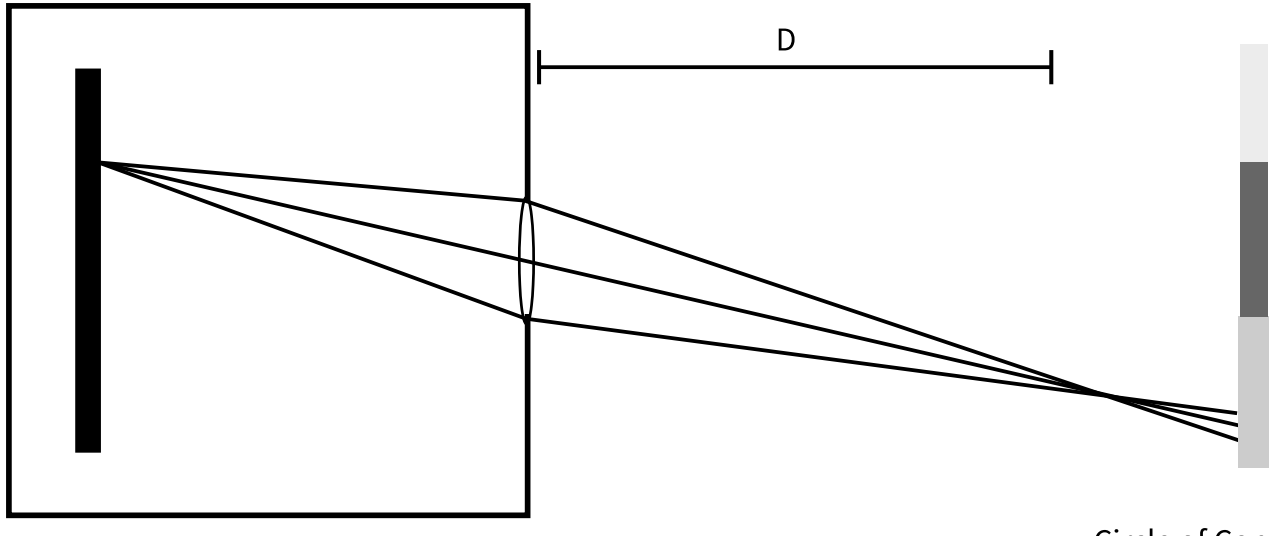
More light & localization of rays to sensor plane
but only for objects at the focusing distance D.

LENSES



Again, point projects to
'circle of confusion'

LENSES



Circle of Confusion still
Proportional to Aperture Size

Can still tradeoff amount of light with amount of blur in out of focus regions (move up or down "f-stops")

Better trade-off than pinhole camera

TRADEOFFS

Photographers think about these tradeoffs every time they take a shot

- Dynamic range and what part of the image should be well exposed (rounding and clipping)

- Choosing between:
  - ISO i.e. Gain & noise
  - Exposure Time & motion blur
  - F-stop i.e. aperture size & defocus blur

COLOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q\ dx\ dy \right] dt$$

We left out at an important term in this equation: wavelength

COLOR

$$I^0[n_x, n_y] = \int_{t=0}^{T} \left[\int E(\lambda, x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ q(\lambda)\ d\lambda\ dx\ dy \right] dt$$

- Light carries different amounts of power in different wavelengths
- $E(\lambda, x, y, t)$ now refers to power per unit area per unit wavelength
  - In wavlength λ, incident at (x, y) at time t
  - Both spectral and spatial density function
- $q(\lambda)$: Quantum efficiency also a function of wavelength
  - CMOS/CCD sensors are sensitive (have high q) across most of the visible spectrum
  - Actually extend to longer than visible wavelengths (near infra red)
  - Why cameras have NIR filter, to prevent NIR radiation from being 'superimposed' on the image

Q: But this measures 'total' power in all wavelengths. How do we measure color ?

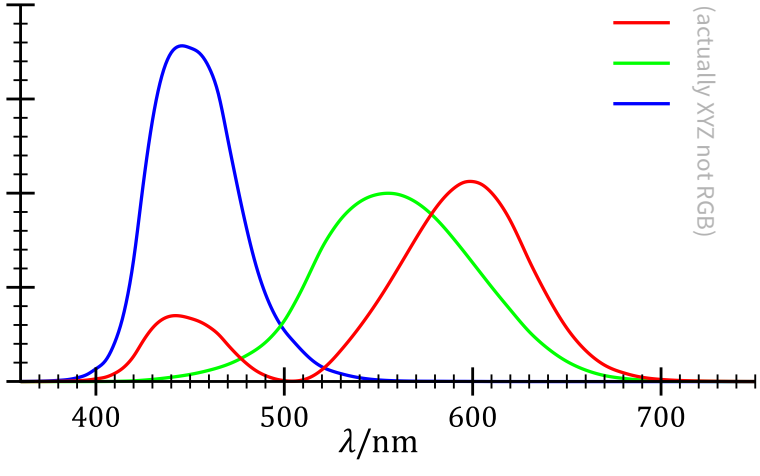
Ans: By putting a color filter in front of each sensor element.

COLOR

$$I^0[n_x, n_y, c] = \int_{t=0}^{T} \left[\int E(\lambda, x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ \Pi_c(\lambda)\ q(\lambda)\ d\lambda\ dx\ dy \right] dt$$

$$\text{for} \quad c \in \{R, G, B\}$$

- Π_c is the transmittance of a color filter for color channel c
- E.g, Π_R will transmit power in (be high for) wavelengths in the red part of the visible spectrum and attenuate power in (be low for) other wavelengths.
- Sometimes also called "color matching functions"

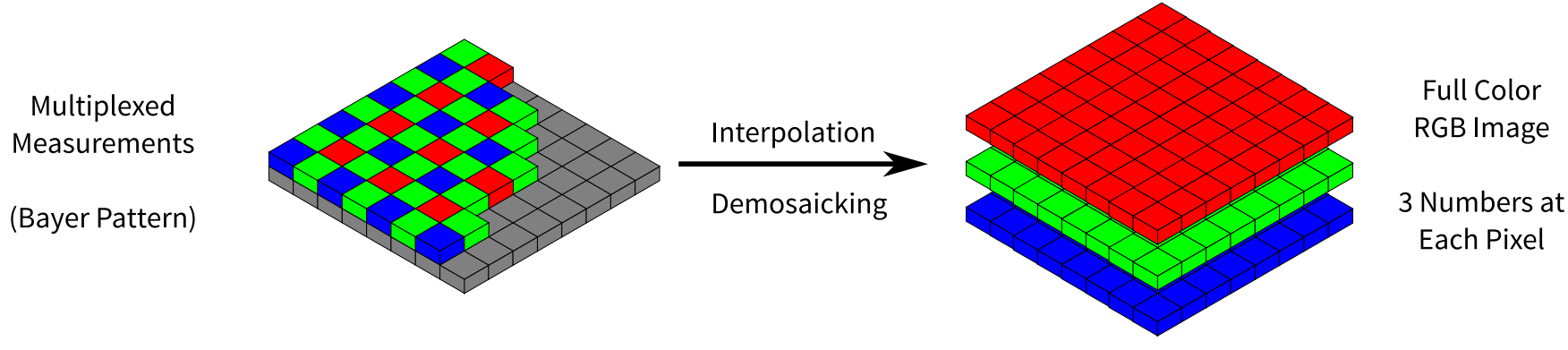


COLOR

$$I^0[n_x, n_y, c] = \int_{t=0}^{T} \left[\int E(\lambda, x, y, t)\ \ p(x - \bar{x}_{n_x}, y - \bar{y}_{n_y})\ \Pi_c(\lambda)\ q(\lambda)\ d\lambda\ dx\ dy \right] dt$$

$$\text{for} \quad c \in \{R, G, B\}$$

- But we can only put one filter in front of each sensor element / pixel location.
- So color cameras "multiplex" color measurements: they measure a different color channel at each location.
- Usually in an alternating pattern called the Bayer pattern:



- Note: a disadvantage is that color filters block light, so measured I^0 values are lower.
- That's why black and white / grayscale cameras are "faster" than color cameras.

FINAL STEPS

Final steps in camera processing pipelines (except for some DSLR cameras shooting in RAW):

- Filter Colors to Standard RGB:
  - Cameras often use their own color filters Π_c.
  - Apply a linear transformation to map those measurements to standard RGB.
- White-balance: scale color channels to remove color cast from a non-neutral illuminant.
- Tone-mapping:
  - The simplest form is "gamma correction" (approximately raising each intensity to the power $(1/2.2)$)
  - Done based on standard developed for what old display devices expected
  - Fits the full set of measurable colors into the gamut that can be displayed / printed
  - Modern cameras often do more advanced processing (to make colors look vibrant)
- Compression

And that's how you get your PNG / JPEG images !

Optional Additional Reading: Szeliski Sec 2.3

OTHER EFFECTS

Other effects we did not talk about. E.g.,

- Real Lenses not thin lenses and have distortions:

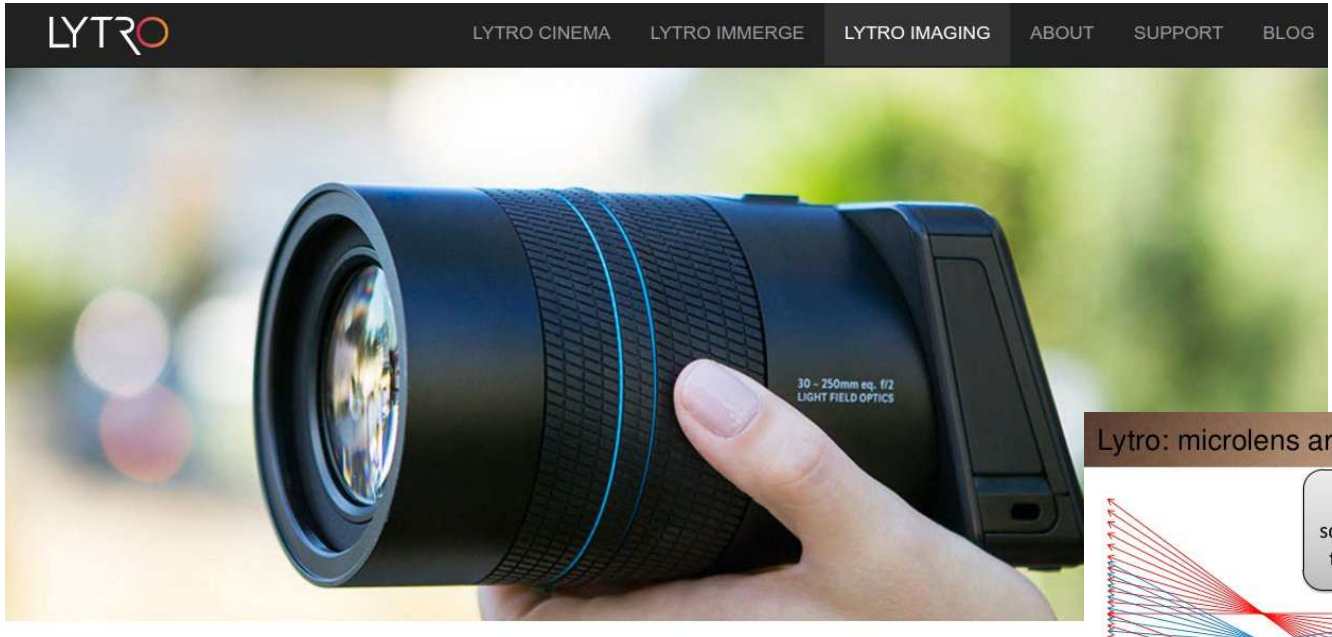
Radial Distortion

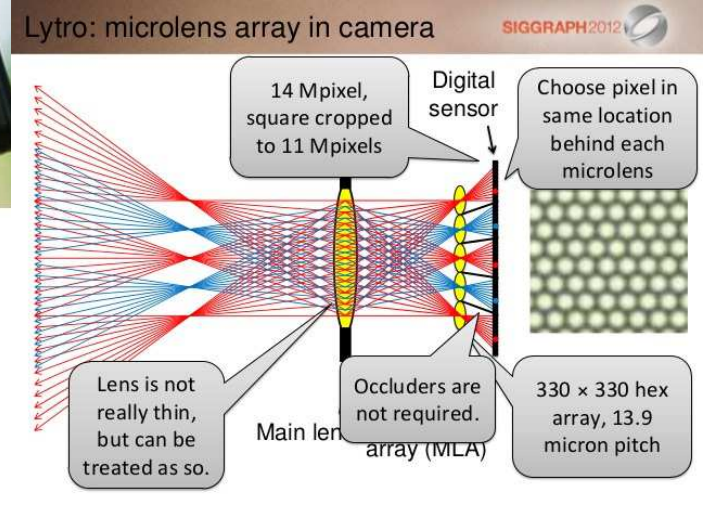
Vignetting

Chromatic Aberration

- Rolling Shutter: No explicit shutter but when pixels reset electronically (along scanlines)

NON-STANDARD CAMERAS





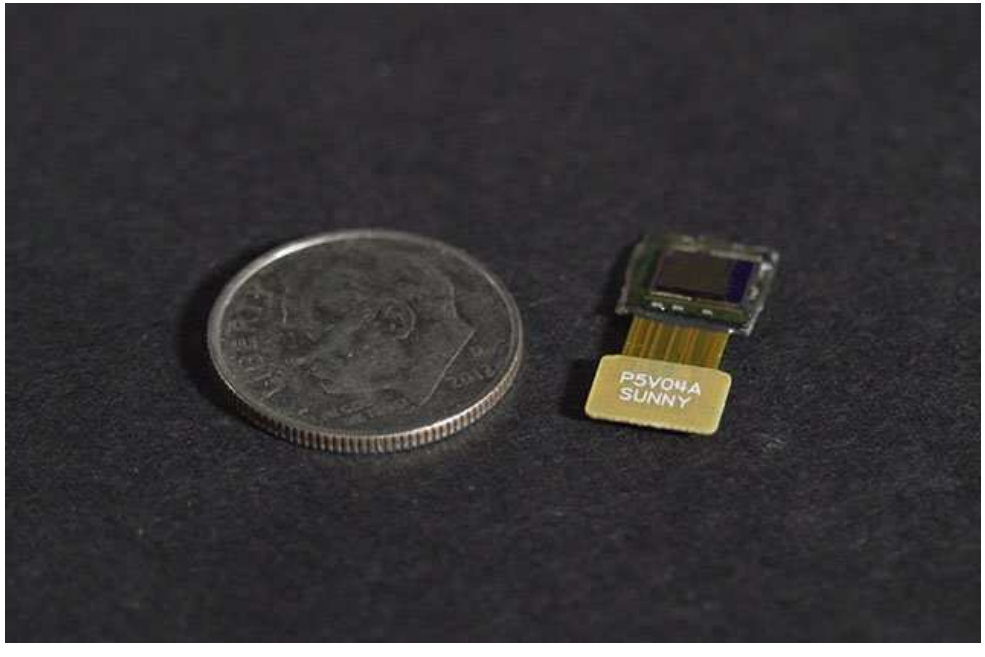
Place array of micro-lenses infront of sensor
Different pixels observe the scene "from a different angle"
Let you estimate depth, shift view point post-capture,
refocus post-capture

NON-STANDARD CAMERAS

16 Different Camera Units

Fuse images to get lower noise, higher dynamic range, synthetically control focal distance and aperture size post-capture

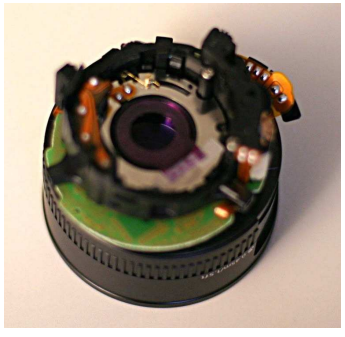
NON-STANDARD CAMERAS



M.S. Asif, A. Ayremolu, A. Sankaranarayanan, A. Veeraraghavan, R. Baranuik,
"FlatCam: Thin, Bare-Sensor Cameras using Coded Aperture and Computation", 2015.

No lens: lets the camera be much smaller, sensor can be placed on curved surfaces.
Put a mask in the aperture: causes 'patterns' of confusion instead of circles of confusion.
Create focused image computationally.

NON-STANDARD CAMERAS



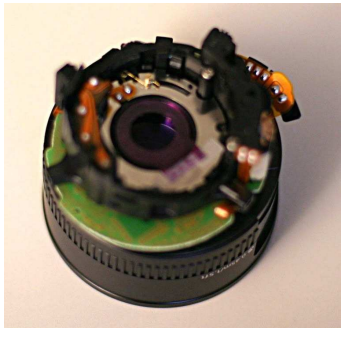
All Focus Image

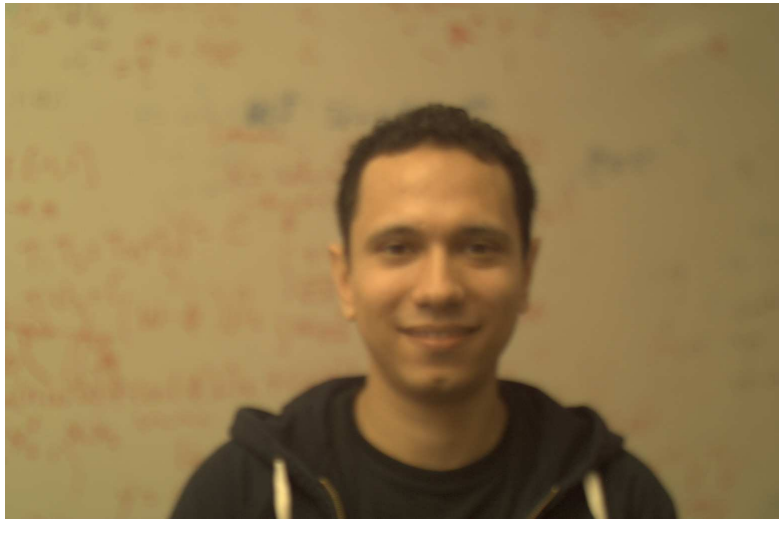
Estimated Depth Map

Coded Aperture Photography

E.g., A. Chakrabarti, T. Zickler. "Depth and Deblurring from a Spectrally-varying Depth-of-Field," ECCV 2012

NON-STANDARD CAMERAS



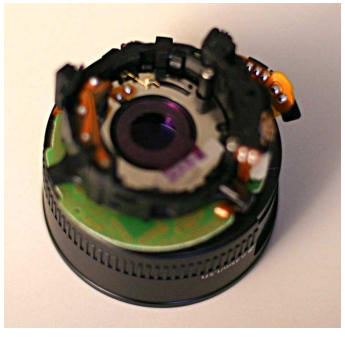


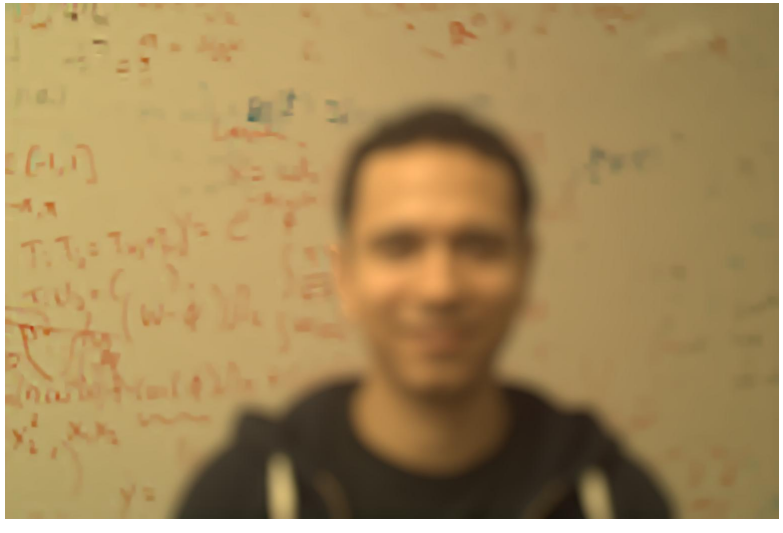
Synthetic Focus Post-capture

Coded Aperture Photography

E.g., A. Chakrabarti, T. Zickler. "Depth and Deblurring from a Spectrally-varying Depth-of-Field," ECCV 2012

NON-STANDARD CAMERAS





Synthetic Focus Post-capture

Coded Aperture Photography

E.g., A. Chakrabarti, T. Zickler. "Depth and Deblurring from a Spectrally-varying Depth-of-Field," ECCV 2012

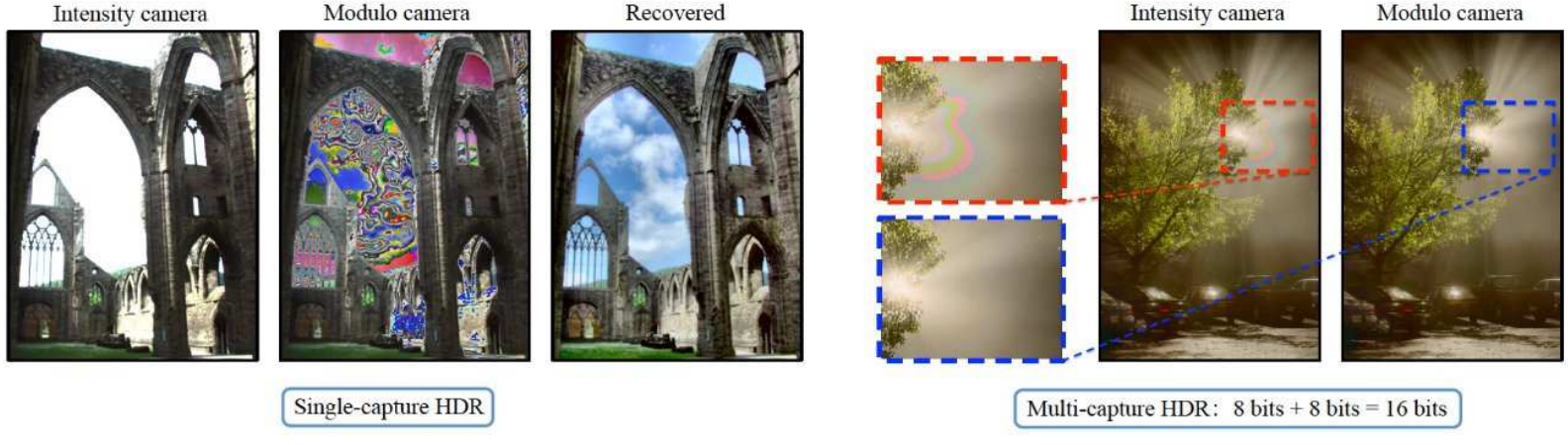
NON-STANDARD CAMERAS

Unbounded High Dynamic Range Photography using a Modulo Camera

Hang Zhao[1] Boxin Shi[1,3] Christy Fernandez-Cull[2] Sai-Kit Yeung[3] Ramesh Raskar[1]

[1] MIT Media Lab [2]MIT Lincoln Lab [3]SUTD

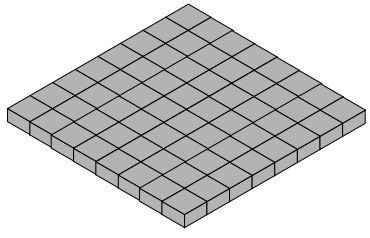
ICCP 2015, Houston, TX **[Best Paper runner-up]**

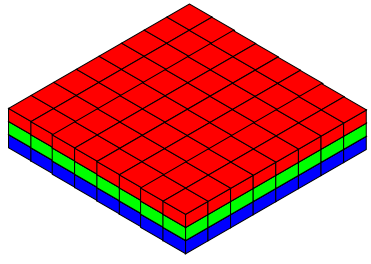


IMAGES

- Exist as 2-D (grayscale) or 3-D (color image) arrays

W x H





W x H x 3

[numpy order: (H, W) or (H,W,3)]

- Precision: uint8 (0-255), uint16(0-65535), Floating point
  - We will often treat them as (positive) real numbers.
- Conventions: $I[n_x, n_y, c], I[n_x, n_y] \in \mathbb{R}$ or $\in \mathbb{R}^3$, $I[n]$: where $n \in \mathbb{Z}^2$

- How do you process / manipulate these arrays ?

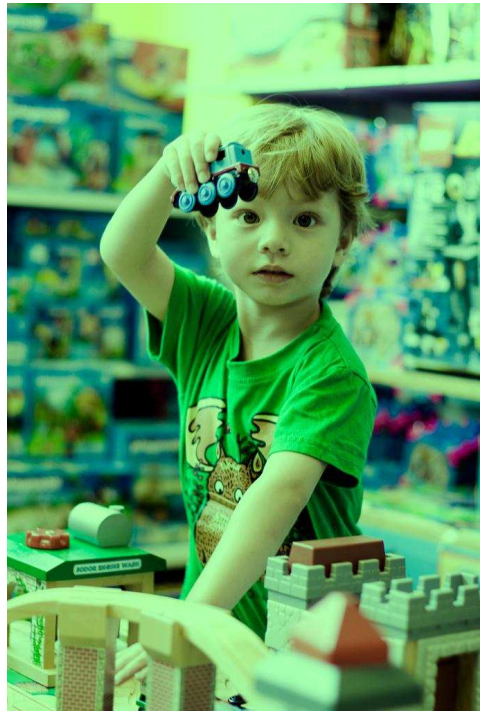
POINT-WISE OPERATIONS

- $Y[n] = h(X[n])$
- $Y[n] = h(X_1[n], X_2[n], \ldots))$
- $Y[n] = h_n(X[n])$ - Might vary based on location.
- $h(\cdot)$ itself might be based on 'global statistics'

POINT-WISE OPERATIONS

[Credit: horizontal.integration @ flickr]

$$X[n] \in \mathbb{R}^3$$



$$Y[n] = \begin{bmatrix} 0.7 & & \\ & 1.05 & \\ & & 0.7 \end{bmatrix} X[n]$$

Linear Color Transforms

POINT-WISE OPERATIONS

$$X[n] \in \mathbb{R}^3$$

$$Y[n] = \begin{bmatrix} 0.7 & & \\ & 0.7 & \\ & & 1.05 \end{bmatrix} X[n]$$

Linear Color Transforms

POINT-WISE OPERATIONS

$$X[n] \in \mathbb{R}^3$$

$$Y[n] = \begin{bmatrix} 1.05 & & \\ & 0.7 & \\ & & 0.7 \end{bmatrix} X[n]$$

Linear Color Transforms

POINT-WISE OPERATIONS

$X[n] \in \mathbb{R}^3$

$$Y[n] = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} X[n]$$

Linear Color Transforms

POINT-WISE OPERATIONS

$$X[n] \in \mathbb{R}^3$$

$$Y[n] = 0.9 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} X[n] + 0.1 \quad X[n]$$

Linear Color Transforms

POINT-WISE OPERATIONS

$X[n] \in \mathbb{R}^3$

$$Y[n] = 0.5 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} X[n] + 0.5 \; X[n]$$

Linear Color Transforms

POINT-WISE OPERATIONS

$X[n] \in \mathbb{R}^3$

$$Y[n] = \text{-1.0} \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} X[n] + 2.0 \quad X[n]$$

Linear Color Transforms

POINT-WISE OPERATIONS

$$X[n] \in \mathbb{R}$$

$$Y[n] = X[n]^{0.5}$$

Non-linear Transforms

POINT-WISE OPERATIONS

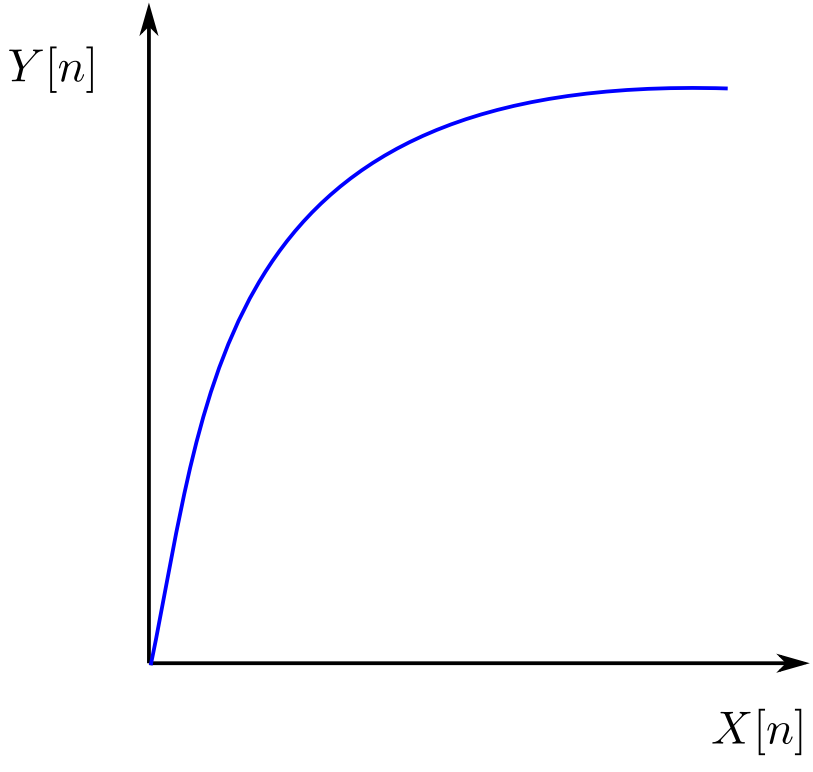
$$X[n] \in \mathbb{R}$$

$$Y[n] = X[n]^{2.0}$$

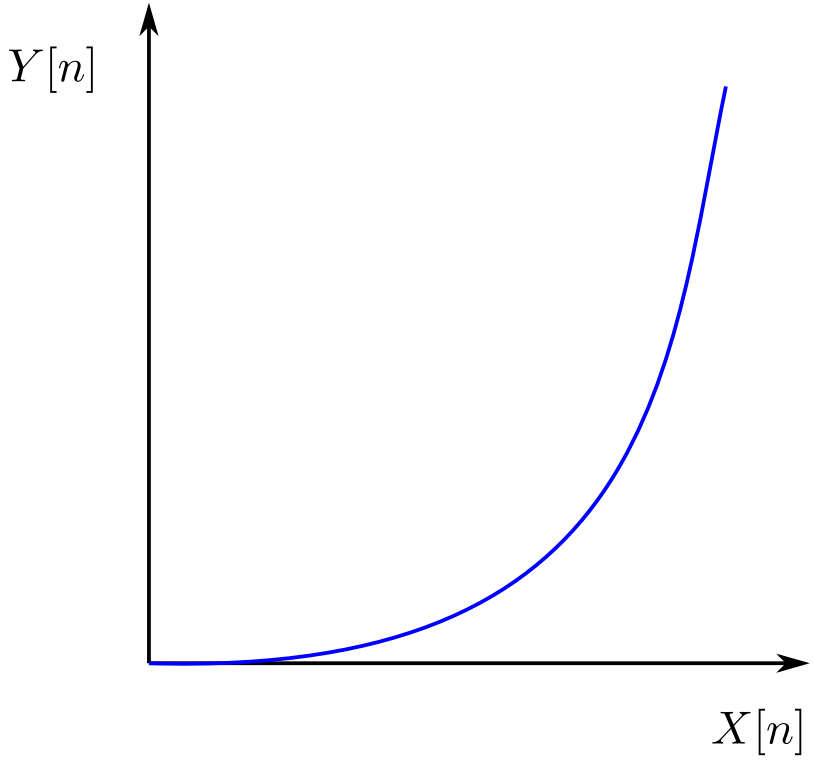
Non-linear Transforms

Tone-maps / Change contrast

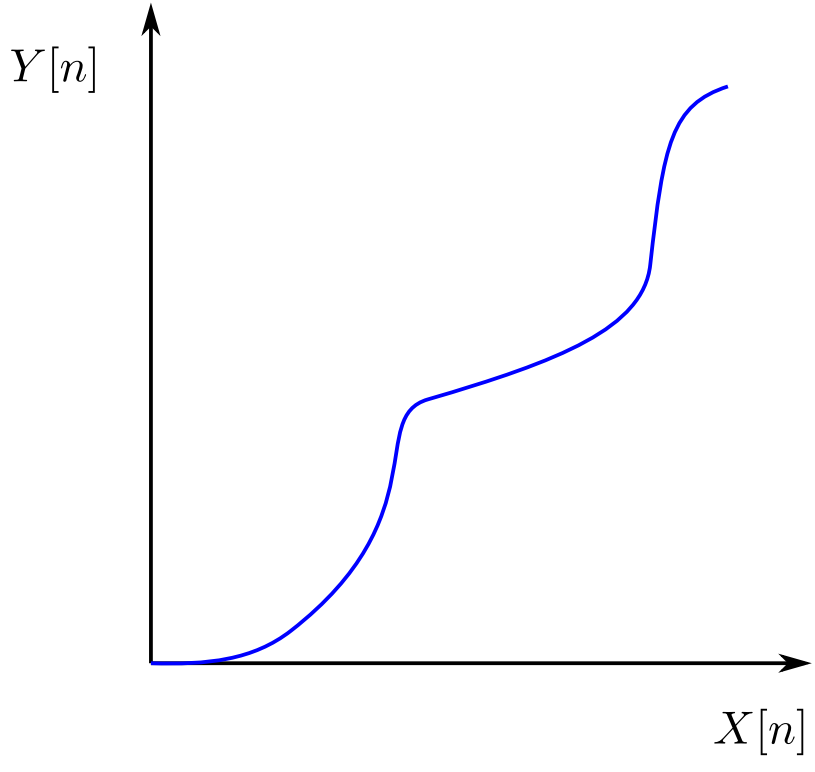
POINT-WISE OPERATIONS



POINT-WISE OPERATIONS



POINT-WISE OPERATIONS

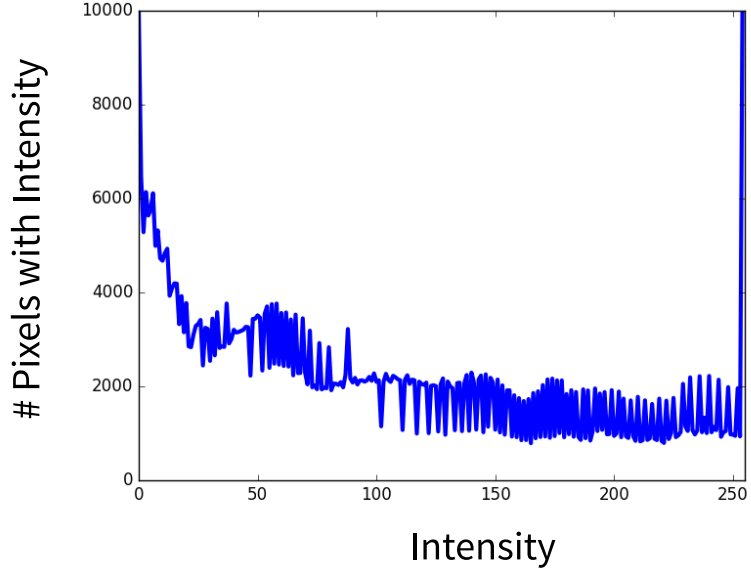


Can be arbitrary: but usually monotonic

If $X[n_1] > X[n_2]$, $Y[n_1] \geq Y[n_2]$.

POINT-WISE OPERATIONS

$X[n]$



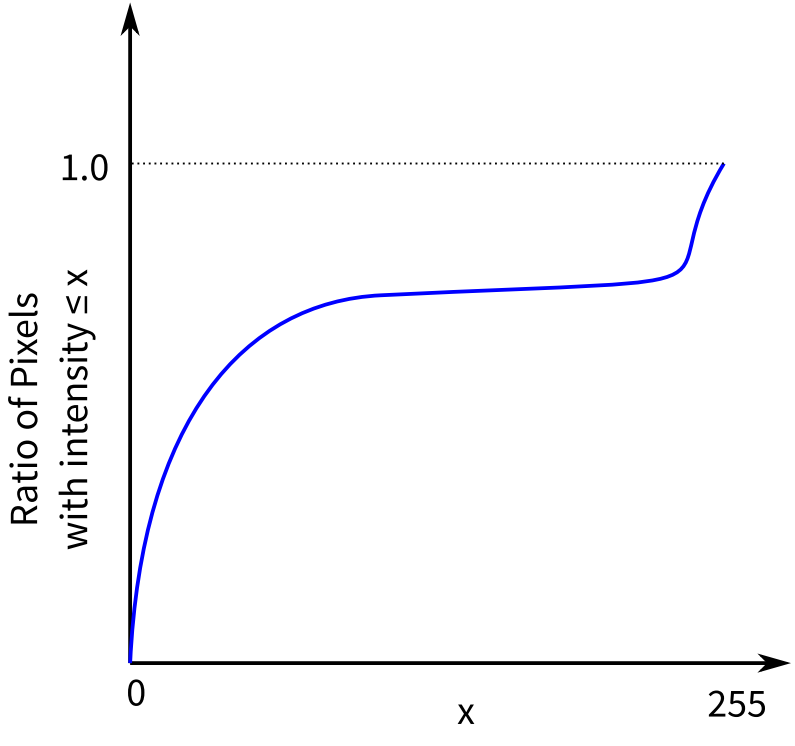
Uneven distribution of intensities (many too dark or too bright)

Find a monotonic function $h(\cdot)$ such that the intensities of $h(X[n])$ are distributed uniformly in the range $[0, 255]$.

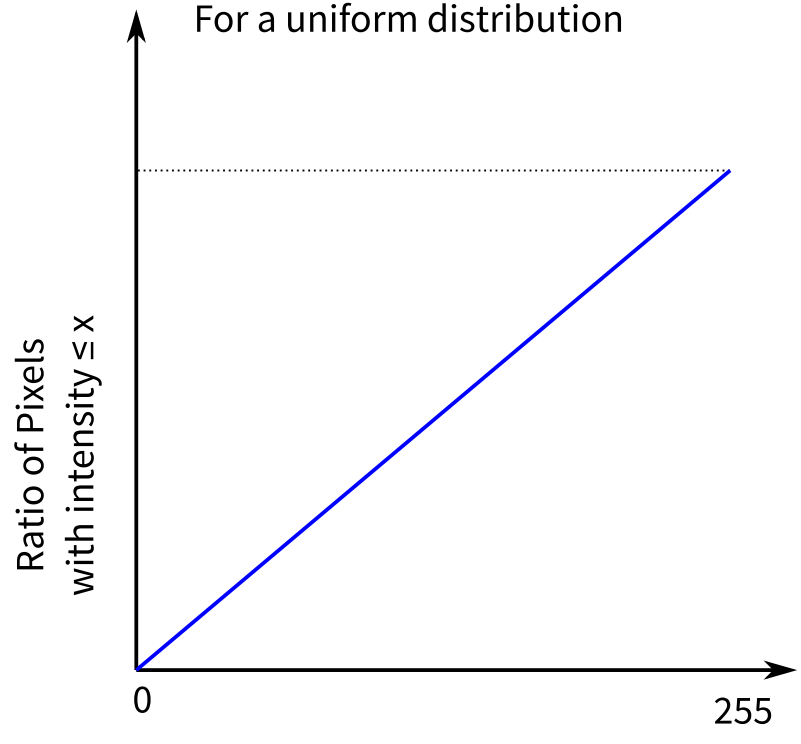
Histogram Equalization

POINT-WISE OPERATIONS

Consider the Cumulative Distribution (think of intensity as a continuous r.v.)

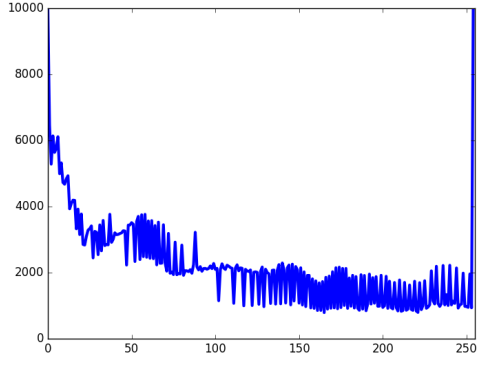


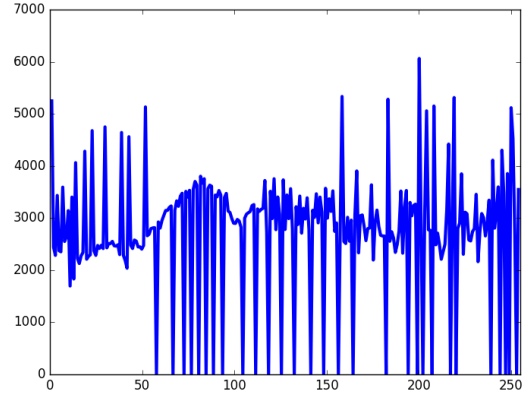
For a uniform distribution



Use the CDF as the tone map: $h(x) = P(X[n] < x) \times 255.0$

POINT-WISE OPERATIONS





Not perfect as dealing with quantized values

POINT-WISE OPERATIONS

Image Matting (combine multiple images with alpha matte)

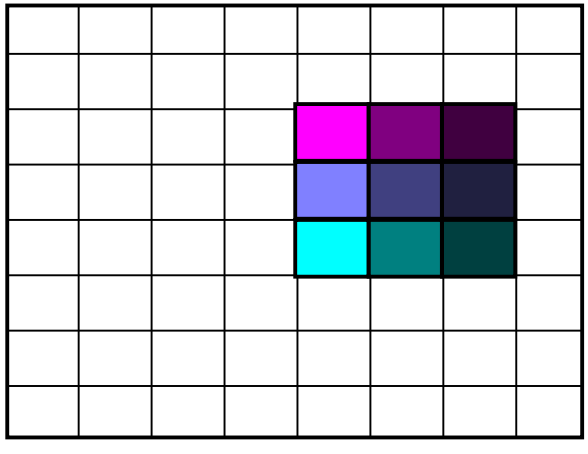
From Szeliski 3.1

CONVOLUTION

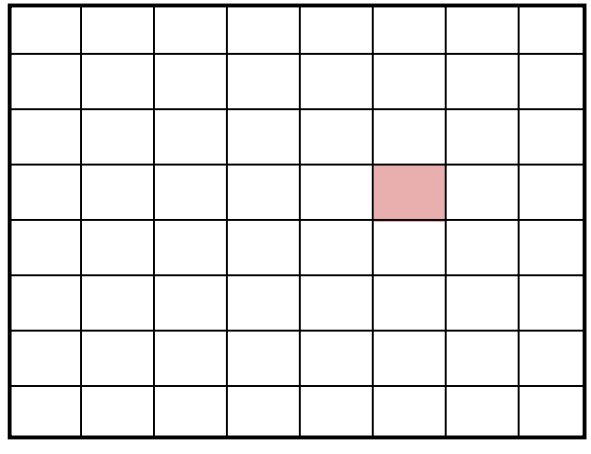
Linear operation on spatial neighborhoods

$$Y[n] = \sum_{n'} X[n - n']k[n']$$

Kernel



$X[n]$



$Y[n]$

$k[n]$

CONVOLUTION

CONVOLUTION

CONVOLUTION